

الجامعة العربية الخاصة للعلوم والتكنولوجيا

كلية الهندسة المعلوماتيّة

السنة الخامسة – الفصل الدراسي الأول

النظم الخبيرة (٢)
Expert Systems

م.عبادة عثمان آغا

م.طلال سلطان

قاعدة المعرفة (مثال عملي)

مثال عملي:

ولتوضيح الفكرة نفترض أننا جمعنا المعرفة حول بعض قواعد الأمان في المختبرات الكيمائية والتي كانت تمثل الضوابط التالية:

قاعدة 1: إذا سكبت مادة حارقة => استدعي الحريق.

قاعدة 2:إذا كان التركيز الكيمائي للمادة أقل من 6 => فان المادة تمثل حامضا.

قاعدة 3:إذا كانت المادة المسكوبة حامضا و رائحتها مثل الخل => فإنها حامض الخليك.

قاعدة 4: إذا كانت المادة حامض الخليك => فإنها مادة غير حارقة.

قاعدة المعرفة (مثال عملي)

وإذا سكب احدهم مادة توفرت عنـها الحقائق التاليـة:

قاعدة 1: إذا سكبت مادة حارقة => استدعي الحريق.

قاعدة 2:إذا كان التركيز الكيمائي للمادة أقل من 6 => فان المادة تمثل حامضا.

قاعدة 3:إذا كانت المادة المسكوبة حامضا و رائحتها مثل الخل => فإنـها حامض الخليك.

قاعدة 4: إذا كانت المادة حامض الخليك => فإنـها مادة غير سامة.

حقيقة 1 : أن تركيزها الكيمائي أقل من 6

حقيقة ٢ : أن رائحتها تشبه الخل.

قاعدة المعرفة (مثال عملي)

فعند مقارنة الحقائق بالقواعد حيث نجد مطابقة بين الحقيقة 1 والقاعدة 2 معا ينتج عن ذلك حقيقة جديدة هي:

حقيقة 3 : أن المادة هي حامض.

وعند إعادة عملية المقارنة نجد تطابقا بين الحقائق2 و 3 والقاعدة ٣ معا ينتج حقيقة جديدة هي:

حقيقة 4 : أن المادة هي حامض الخليك.

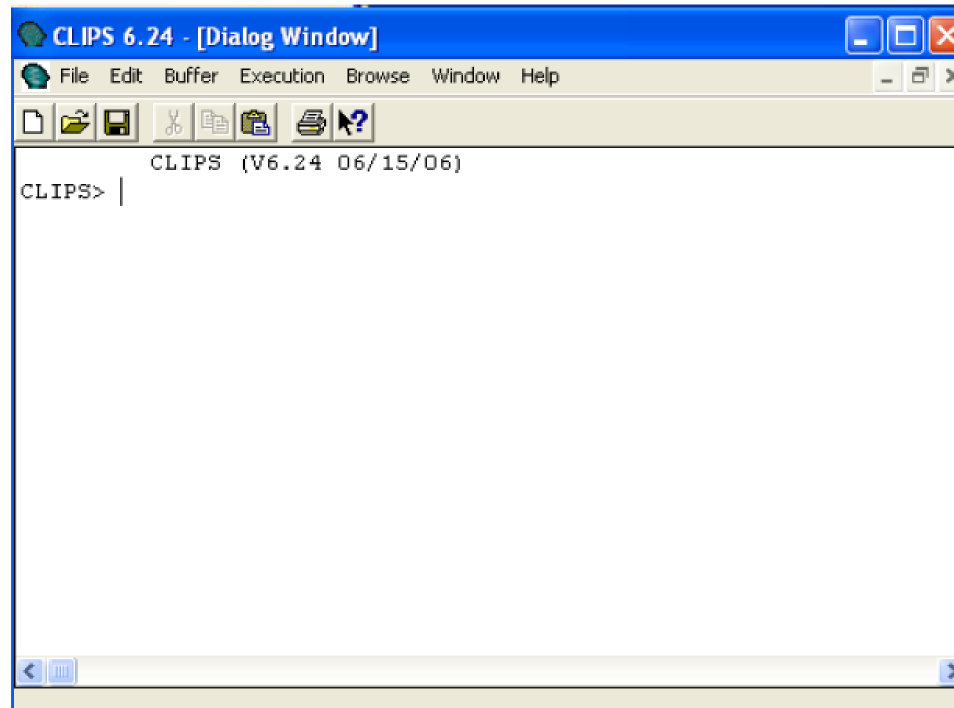
وبنفس الطريقة تتولد لنا حقيقة خامسة جديدة هي:

حقيقة 5 :أن المادة ليست حارقة.

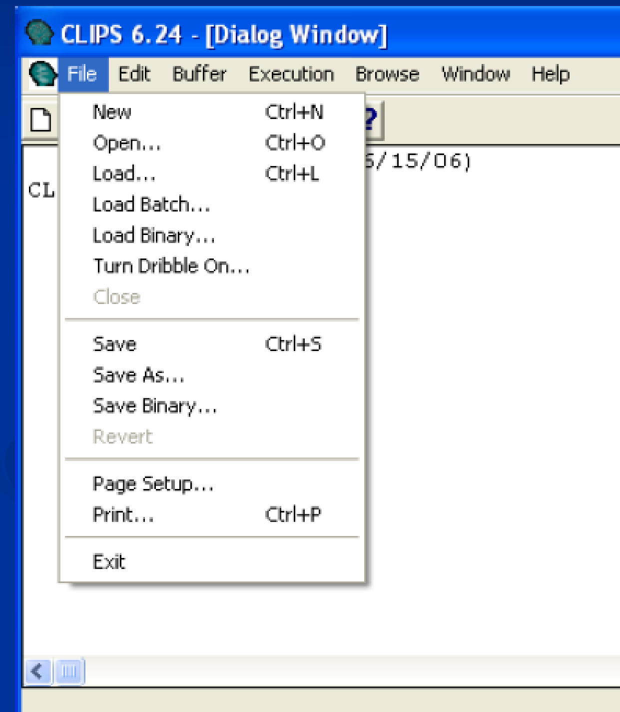
أي أن ذلك لا يشكل خطرا يستدعي مناداة الحريق.

CLIPS

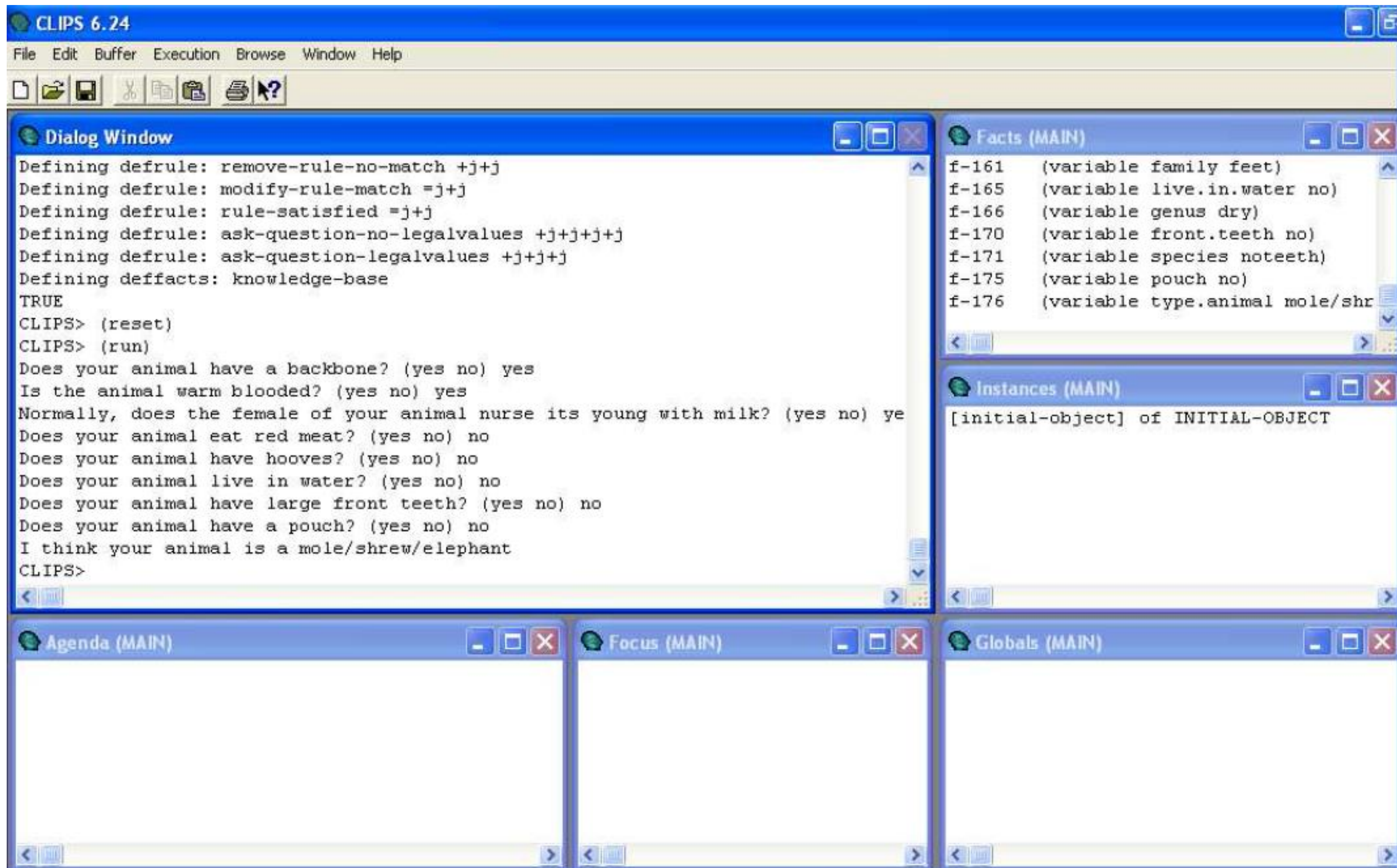
- Clips هو اختصار لـ **C Language Integrated Production System**
- وهـو أداة لبنـاء الأنظمـة الخبيـرة، كمـا أنـه يُصـنّف كلغـة برمجيـة متعـددة النمـاذج (multi paradigm programming language) تؤمن دعم كبير للبرمجـة التـي تعتمـد علـى القواعد والبرمجـة كائنيـة التوجـه (rule-based & object-oriented programming).



- Select **File** -> **Load** to load a clips program
- Select **Execution** -> **Reset** to reset memory
- Select **Execution** -> **Run** to execute the program
- Select **Execution** -> **Clear CLIPS** to clear the memory
- Select **Browse** -> **Deffacts Manager** to list the name of deffacts construct
- Select **Browse** -> **Deftemplate Manager** to list the name of Deftemplate construct
- Select **Window** -> **Facts** to list the facts in memory
- Select **Window** -> **Agenda** to list the rule to be fired at the current step
- Select **Window** -> **Globals** Window to list the globals variance



قائمة بأهم الأوامر التنفيذية في Clips



نوافذ clips (توجد ضمن تبويبة Window)

العناصر الأساسية في النظام الخبير هي:

- قائمة حقائق (Fact-List): وتُصنف على أنها ذاكرة عامة للبيانات (Global memory for data).
- قاعدة معرفة (Knowledge-base): وتتضمن القواعد.
- محرك استدلال (Inference Engine): والذي يتحكم بالتنفيذ ككل.

تتألف البرمجة في clips بشكل رئيسي من:

- حقائق (Facts)
- قواعد (rules)

- يظهر مؤشر clips كالآتي:

CLIPS>

- يُمكن إدخال الأوامر بشكل مباشر إلى clips ، ويُدعى هذا المستوى بـ top level.
- كما يُمكن الخروج باستخدام (exit).
- كل تعليمة برمجية يجب أن يكون لها عدد متساوي من الأقواس اليسارية واليمينية، وإلا سينتج خطأ.

CLIPS Programming

Overview

(exit) shutdown Clips

(clear) Remove all rules and fact from memory

(reset) Removes facts information from memory

(run) start executing Clips programming

CLIPS Programming

Basics

- **Numbers:**

  5 7. 5.693

- **Strings:**

  “too " "Hello, World?! "

CLIPS Programming

Basics

- **Lists**

```
(a b c)

(+ 3 2)

(deftemplate person (slot name) (slot age))
```

- **Comments**

```
; This is a list  (a b c)
```

CLIPS Programming

Functions

Clips> (+ 2 3)

Clips> (+ (+ 2 3) (* 3 3))

Clips> (printout t "The number is “ 8 "!" crlf)

//create$ function appends any number of fields together to create a multifield value.

```
CLIPS (create$ hammer drill saw screw pliers wrench)
(hammer drill saw screw pliers wrench)
CLIPS> (create$ (+ 3 4) (* 2 3) (/ 8 4))
(7 6 2)
CLIPS>
```

التعامل مع أنواع البيانات يشابه بقية اللغات البرمجية الأخرى

```
; both integer arguments
; give integer result
CLIPS> (+ 2 2)
4
; both floating-point arguments
; give floating-point result
CLIPS> (+ 2.0 2.0)
4.0
; mixed arguments
; give floating-point result
CLIPS> (+ 2 2.0)
 4.0

; convert integer
; to float
CLIPS> (float (+ 2 2))
4.0
; convert float
; to integer
CLIPS> (integer (+ 2.0 2.0))
4
CLIPS>
```

CLIPS Programming

Variables

- (bind **?x** “my name is ahmad“)
- (bind **?b** 59)

- ***//multivariable with $......try***
- (bind **?breakfast-list** (create$ eggs bread tea))

- ***//to see the value of a variable.....try***

?a

CLIPS Programming

Global variables (or defglobals)

- (defglobal [?<global-name> = <value>]+)

?*b* ?*counter*

- (defglobal ?*x* = 91)

CLIPS Programming

Global variables (or defglobals)

- not destroyed by reset i.e. :

- **Clips>** (defglobal ?*x* = 3)
- **Clips>** (bind ?*x* 4)
- 4
- **Clips>** (reset)
- TRUE
- **Clips>** ?*x*
- 3

CLIPS Programming

Clips> (< ?a ?b)

TRUE

Clips> (if (< ?a ?b) then (bind ?a 7))

CLIPS Programming

- **Deffunctions**

CLIPS allow You to define your own functions with **deffunction**

- (deffunction <function-name> (<parameter>*)

  <expr>* [<return-specifier>])

- Clips> **(deffunction MaxNum (?a ?b)**

  ;Write function implementation here

)

التوابع Functions

مثل بقية لغات البرمجة، يسمح clips بتعريف التوابع الخاصة بنا باستخدام deffunction، مما يوفر الجهد بإعادة ادخال نفس الأوامر عدة مرات.

يمكن استدعاء deffunctions بشكل مشابه للاستدعاء الطبيعي للتوابع، كما يُمكن استخدام الـ deffunctions على شكل argument لتابع آخر.

كما يُمكن استخدام printout في أي مكان ضمن الـ deffunction حتى لو لم تكن الإجراء الأخير ضمنها، لأن أمر الطباعة يُعتبر تأثير جانبي لاستدعاء تابع الطباعة (printout).

الصيغة العامة:

```
• (deffunction <function-name> (?arg1 ?arg2 ... ?argM [$?argN])
          <expr>*     [<return-specifier>])
```

```
•Clips> (deffunction   MaxNum (?a ?b)
            (if (> ?a ?b)  then       (printout t ?a)
             else (printout t ?b)    ))
```

```
•Clips> (deffunction   MaxNum (?a ?b)
            (if (> ?a ?b)  then       (return ?a)
             else       (return ?b)    ))
```

التوابع Functions

- ?arg هي dummy arguments ، أي أن أسماء الـ arguments لن تتداخل مع أسماء المتغيرات في قاعدة ما إذا كانت متماثلة.

- مع العلم أن مصطلح الـ dummy arguments يُدعى أحياناً parameters في مراجع أخرى.

- عند توفر مجموعة اجراءات actions، وعلى الرغم من أن كل إجراء سيُعيد قيم values من استدعاءات التابع ضمن الإجراء، إلا أنه يتم حظر إعادة هذه القيم للمستخدم من قبل الـ deffunction، حيث يُعيد قيمة الإجراء الأخير فقط، حيث أن هذا الإجراء يُمكن أن يكون تابع أو متغير أو ثابت ..الخ.

مثال لتعريف تابع لحساب وتر المثلث، حيث نجد أن استخدام نفس أسماء المتغيرات في الـ dummy arguments لن يسبب تداخلاً.

```
CLIPS> (clear)
CLIPS>
(deffunction hypotenuse   ; name
   ; dummy arguments
   (?a ?b)
   ; action
   (sqrt(+ (* ?a ?a) (* ?b ?b))))
CLIPS>
```

التوابع Functions

يُمكن استخدام deffunctions وفق قيم متعددة الحقول multifield values كما في المثال التالي:

```
CLIPS> (clear)
CLIPS>
(deffunction count ($?arg)
    (length $?arg))
CLIPS> (count 1 2 3 a duck "quacks")
6
CLIPS>
```

ميزات أخرى مفيدة في clips (Other Features)

round	التقريب إلى أقرب عدد صحيح
integer	إزالة الفواصل العشرية
list-deffunctions	عرض قائمة بجميع التوابع
undeffunction	حذف deffunction إذا لم يكن يُنفذ حالياً ولا يتم استدعاؤه في مكان آخر، وإذا تم تحديد اسم التابع المراد حذفه باستخدام "*" ، فإن ذلك سيتسبب بحذف الكل.
length	عدد الحقول ، أو عدد المحارف في string أو symbol.
nth$	يعيد الحقل المحدد إذا كان موجوداً وإلا فإنه لا يُعيد شيء.
subsetp	يعيد true إذا كانت قيمة متعددة الحقول multifield value هي مجموعة فرعية من قيمة متعددة حقول أخرى، وإلا فإنه يُعيد false.

تمرين

1) Write deffunction to find the area of triangle?

Hint: Area formula = (base * hight)/2